

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Masahiko NAKAHARA et al.  
Application No.: To Be Assigned  
Filing Date: April 20, 2001  
Title: PROCESS EXECUTING METHOD AND RESOURCE  
ACCESSING METHOD IN COMPUTER SYSTEM  
Art Unit: 2151  
Examiner: M. BANANKHAH

**PRELIMINARY AMENDMENT**

Assistant Commissioner of Patents  
Washington, D.C. 20231

April 20, 2001

Sir:

Prior to examination, please enter thereto the following Preliminary Amendment.

**IN THE SPECIFICATION:**

Please amend the specification as follows:

**Page 1**, following the title, please **insert** the following paragraph:

**-- Cross-Reference to Related Application**

This application is a Divisional of U.S. Application Number 08/917,477, filed August 26, 1997.--;

Page 2, line 16, change "an" to -a-;  
Page 7, line 7, delete "the" at first occurrence;  
Page 8, line 13, change "whereon" to -wherein-;  
line 27, delete "the";

Page 9, line 14,	change "whereon" to -wherein-;
line 18,	delete "the" at first occurrence;
Page 10, line 28,	delete "the" at second occurrence;
Page 11, line 9,	delete "the";
line 19,	delete "the" at first occurrence;
line 22,	change "whereon" to -wherein-;
line 24,	delete "more" to -longer-;
Page 27, line 14,	change "whereon" to -wherein-;
line 21,	delete "the";
line 24,	delete "the" at second and third occurrences;
Page 33, line 16,	delete "the" at first and second occurrences;
line 17,	delete "the" at both occurrences;
line 28,	delete "the" at first occurrence;
Page 41, line 24,	change "whereon" to -wherein-;
Page 47, line 5,	change "whereon" to -wherein-;
line 8,	delete "a" at first occurrence;
line 10,	delete "a";
line 18,	delete "the";
line 24,	change "whereon" to -wherein-;
Page 48, line 17,	change "whereon" to -wherein-;
Page 49, line 17,	change "whereon" to -wherein-;
Page 51, line 24,	change "manage" to -manages-; and
Page 54, line 4,	change "whereon" to -wherein-.

**IN THE CLAIMS:**

**Please cancel claims 1-5 and add claims 6-25, as follows:**

--6. A method of accessing a single resource in an operating system, comprising the steps of:

incrementing a counter included in said operating system by a discrete value, upon generation of a resource, and using a counter value as generation identifying information;

assigning to said resource an identifier composed of address information and the generation identifying information of said resource;

storing said generation identifying information at a leading location of said resource;

extracting generation identifying information from an identifier transferred as an argument of a system call issued by one user application for accessing said resource;

comparing the extracted generation identifying information with the generation identifying information stored in said resource at said leading location; and

enabling access to said resource when coincidence is found between both of said generation identifying information, while disabling access to said resource when discrepancy is found between both of said generation identifying information.

7. An accessing method according to claim 6, wherein said generation identifying information corresponds to a time said resource was generated.

8. An accessing method according to claim 6, wherein said identifier corresponds to a 64-bit identifier containing a 32 most significant bits which indicates the address of said resource and a 32 least significant bits which indicates the generation

identifying information.

9. An accessing method according to claim 6, further comprising:

when said coincidence is found between both the generation identifying information, determining that said identifier of said resource is valid and enabling access to said resource using the resource address to perform the processing of said resource; and

when said discrepancy is found between both the generation identifying information, determining that said identifier of said resource is invalid and disabling access to said resource.

10. An accessing method according to claim 9, wherein said identifier corresponds to a 64-bit identifier containing a 32 most significant bits which indicates the address of said resource and a 32 least significant bits which indicates the generation identifying information.

11. An accessing method according to claim 10, wherein said generation identifying information contains leading 16 bits representing said count value of said counter and trailing 16 bits representing a process identifier.

12. An accessing method according to claim 9, wherein said counter records the number of times said resource is generated, and is set to an initial value of zero and incremented by one each time said resource is generated.

13. A method for controlling access to a shared resource in an operating system

from different processes, comprising the steps of:

incrementing a counter by a discrete value each time a shared resource is requested from a user application for recording a number of times said shared resource is requested from said user application;

assigning to said shared resource requested, a resource identifier comprised of a resource address and generation identifying information assigned to said shared resource;

storing said generation identifying information of said resource identifier at a designated location of said shared resource;

extracting generation identifying information from a resource identifier transferred as an argument of a system call issued by said user application request for accessing said shared resource;

making a comparison between the extracted generation identifying information with the generation identifying information stored at said designated location of said shared resource; and

controlling access to said shared resource in response to said comparison.

**14.** A method according to claim 13, further comprising:

enabling access to said shared resource when the extracted generation identifying information matches the generation identifying information stored at said designated location of said shared resource; and

disabling access to said shared resource when the extracted generation identifying information does not match the generation identifying information stored at said designated location of said shared resource.

15. A method according to claim 14, wherein said shared identifier corresponds to a 64-bit identifier containing a 32 most significant bits which indicates the address of said shared resource and a 32 least significant bits which indicates the generation identifying information.

16. A method according to claim 14, wherein said generation identifying information contains leading 16 bits representing a count value of said counter and trailing 16 bits representing a process identifier.

17. A method according to claim 13, further comprising:  
when the extracted generation identifying information matches the generation identifying information stored at said designated location of said shared resource, determining that said resource identifier of said shared resource is valid and enabling access to said shared resource using the resource address to perform the processing of said shared resource; and

when the extracted generation identifying information does not match the generation identifying information stored at said designated location of said shared resource, determining that said resource identifier of said shared resource is invalid and disabling access to said shared resource.

18. A method according to claim 17, wherein said resource identifier corresponds to a 64-bit identifier containing a 32 most significant bits which indicates the address of said shared resource and a 32 least significant bits which indicates the generation identifying information.

19. A method according to claim 18, wherein said generation identifying information contains leading 16 bits representing a count value of said counter and trailing 16 bits representing a process identifier.

20. A method according to claim 13, wherein said counter records the number of times said shared resource is requested, and is set to an initial value of zero and incremented by one each time said shared resource is requested.

21. A method according to claim 19, wherein said counter records the number of times said shared resource is requested, and is set to an initial value of zero and incremented by one each time said shared resource is requested.

22. A method according to claim 6, further executing a given one of a plurality of processes in said operating system using said resource by:

acquiring said resource for use by said given one process after disabling abortion and preemption of said given one process;

clearing said given one process from preempt-disabled state and disabling preemption of said given one process after processing for said resource;

clearing said given one process from the preempt-disabled state and from the abort-disabled state after said resource has been deallocated from use by said given one process;  
and

executing a forcive termination request issued for said given one process during a period in which said given one process has been in the abort-disabled state.

23. A method according to claim 22, further comprising:  
providing a queue for use of said resource for registering those processes issued respective requests;  
executing in a multiprocessing environment a leading one of the processes registered in said queue and issued respective requests for use of said resource; and  
driving periodically processing relating to said process after completion of execution and executing serially the processes registered in said queue.

24. A method according to claim 13, further executing a given one of a plurality of processes in said operating system using said shared resource by:  
acquiring said shared resource for use by said given one process after disabling abortion and preemption of said given one process;  
clearing said given one process from preempt-disabled state and disabling preemption of said given one process after processing for said shared resource;  
clearing said given one process from the preempt-disabled state and from the abort-disabled state after said shared resource has been deallocated from use by said given one process; and  
executing a forcive termination request issued for said given one process during a period in which said given one process has been in the abort-disabled state.

25. A method according to claim 24, further comprising:  
providing a queue for use of said shared resource for registering those processes issued respective requests;  
executing in a multiprocessing environment a leading one of the processes registered



in said queue and issued respective requests for use of said shared resource; and

driving periodically processing relating to said process after completion of execution and executing serially the processes registered in said queue.--

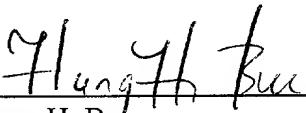
### REMARKS

Claims 6-15 are pending in this application. Originally filed claims 1-5, have been canceled regarding prejudice to avoid duplicity with allowed parent application Serial Number 08/917,477. Claims 6-25 have been added to correspond to previously non-elected claims 3-22 due to Restriction Requirement (Paper No. 6) dated on August 16, 1999 of parent application Serial Number 08/917,477. As pending claims 6-25 are now in condition for further examination. A request for approval of drawing corrections is submitted with this amendment. Upon approval, these revisions will be entered into the formal drawings. Should any questions remain unresolved, the Examiner is requested to telephone Applicants' attorney at (703) 312-6600.

Please charge any shortage in fees due in connection with the filing of this paper, or credit any overpayment of fees, to the deposit account of Antonelli, Terry, Stout & Kraus, LLP, Deposit Account No. 01-2135 (500.35615VX1).

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP

  
\_\_\_\_\_  
Hung H. Bu  
Registration No. 40,415

HHB:srm

1300 North 17<sup>th</sup> St.  
Suite 1800  
Arlington, VA 22209  
Telephone: (703) 312-6600  
Facsimile: (703) 312-6666

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

**Applicants:** Masahiko NAKAHARA et al.  
**Serial No.:** To Be Assigned  
**Filed:** April 20, 2001  
**For:** PROCESS EXECUTING METHOD AND RESOURCE  
ACCESSING METHOD IN COMPUTER SYSTEM  
**Group Art Unit:** 2151  
**Examiner:** M. BANANKHAH

**REQUEST FOR APPROVAL OF DRAWING CORRECTIONS**

Assistant Commissioner for Patents  
Washington, DC 20231

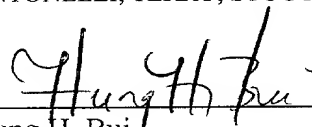
April 20, 2001

Sir:

Approval of the correction to Figures 3, 7, 8, 9, 10, 11, 13, 14, 15, 16, 18, and 24 as indicated on the attached sheets, is respectfully requested.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP

  
Hung H. Bui  
Reg. No. 40,415

(703) 312-6600  
HHB:srm

FIG. 3

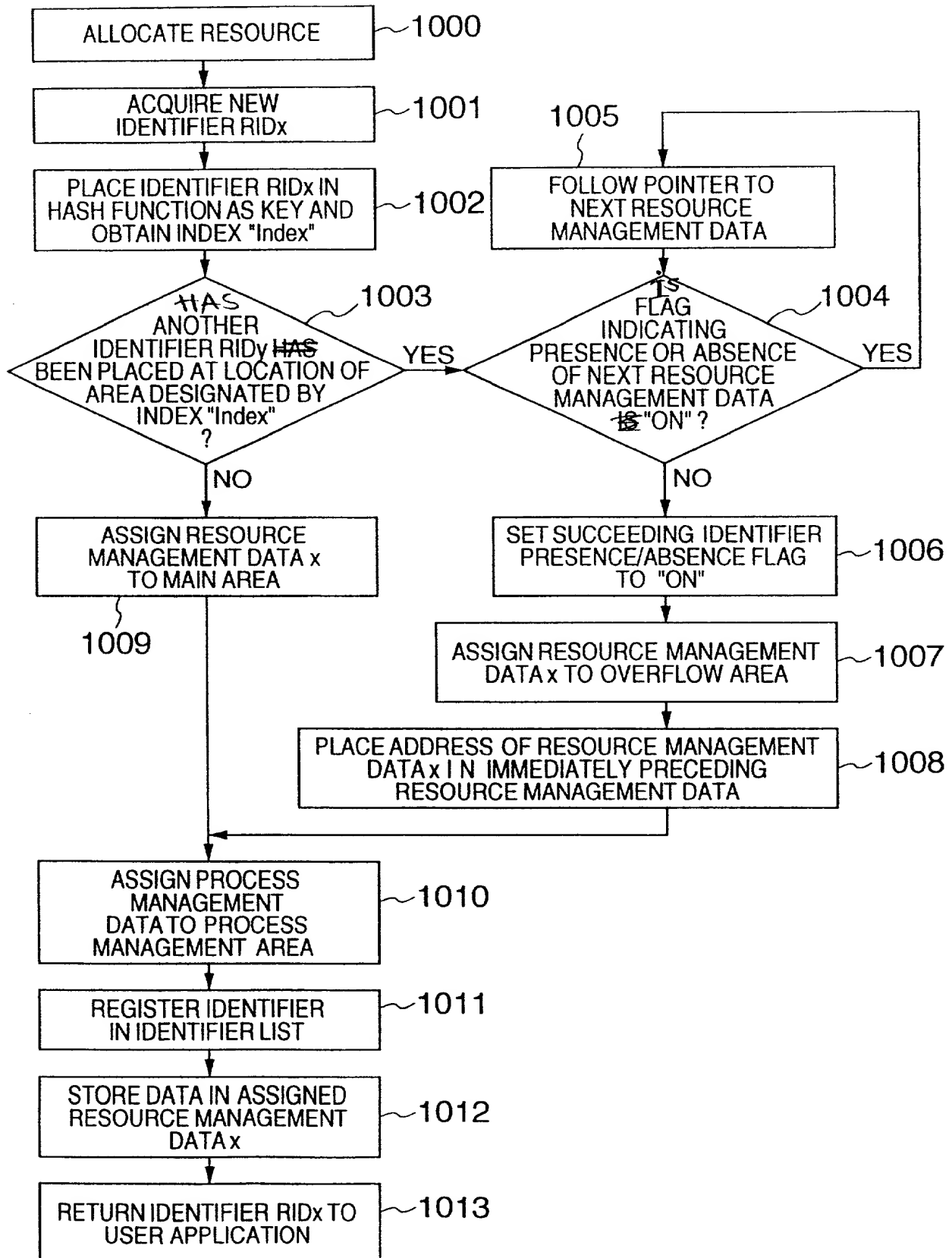
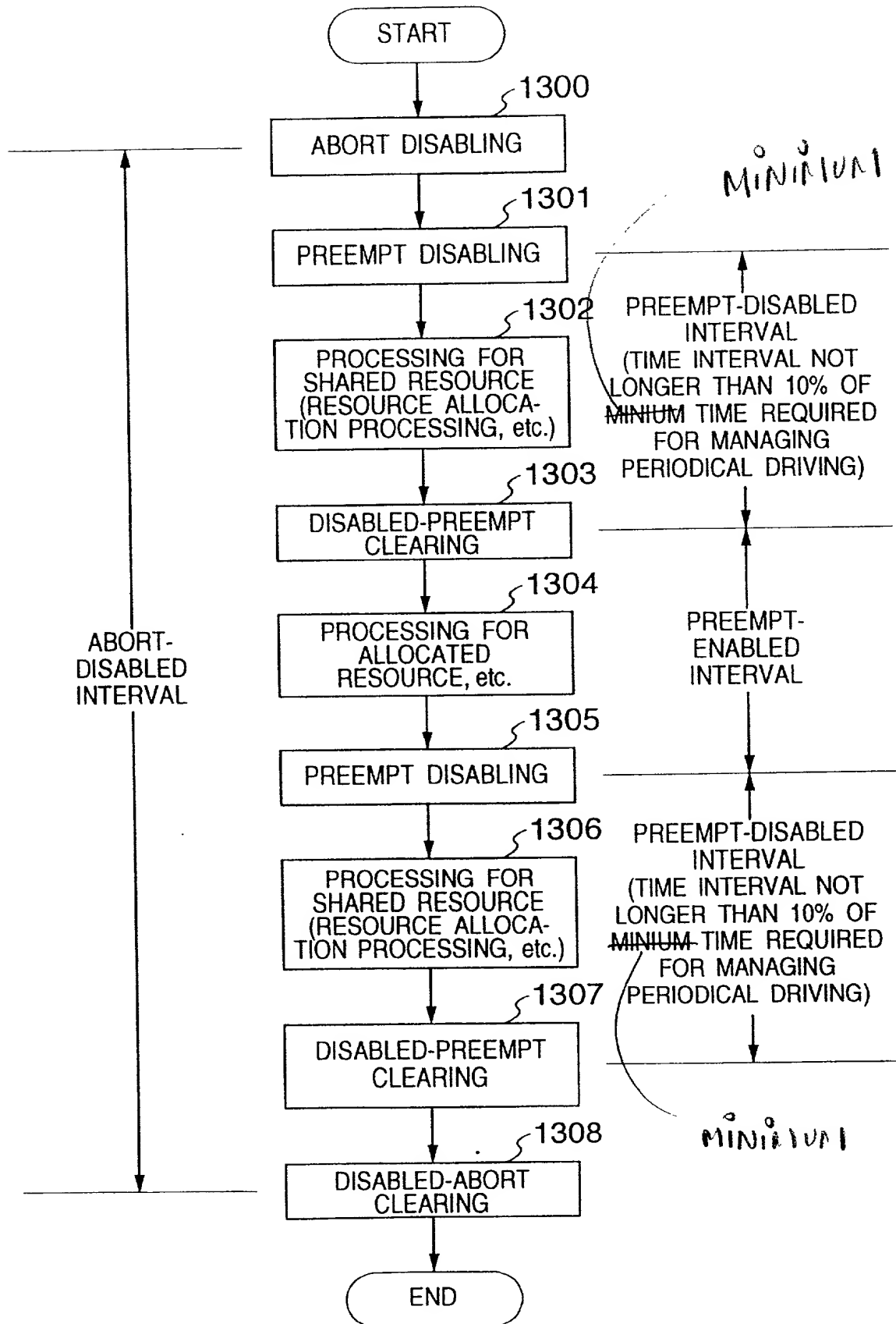


FIG. 7



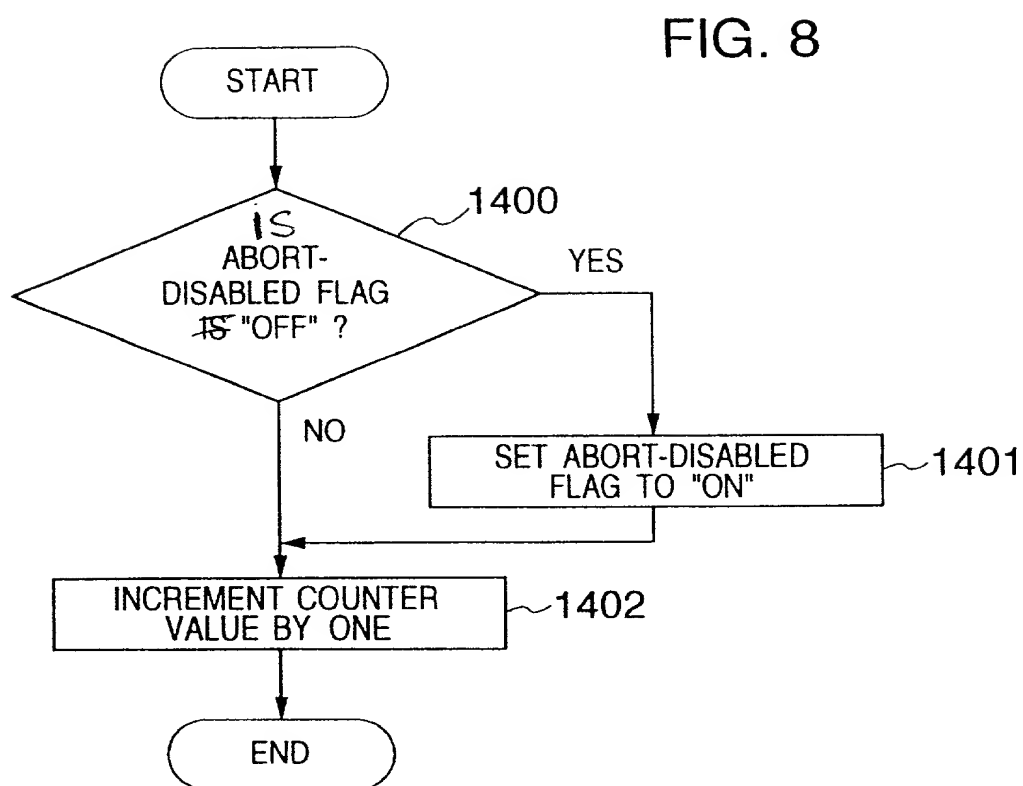


FIG. 9

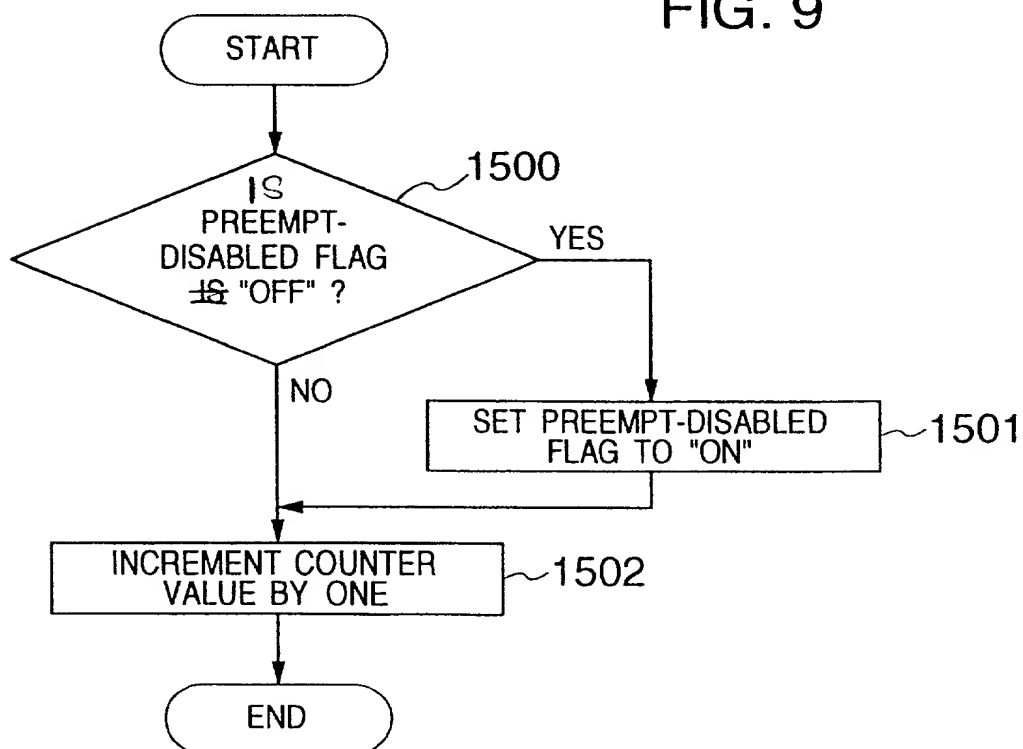


FIG. 10

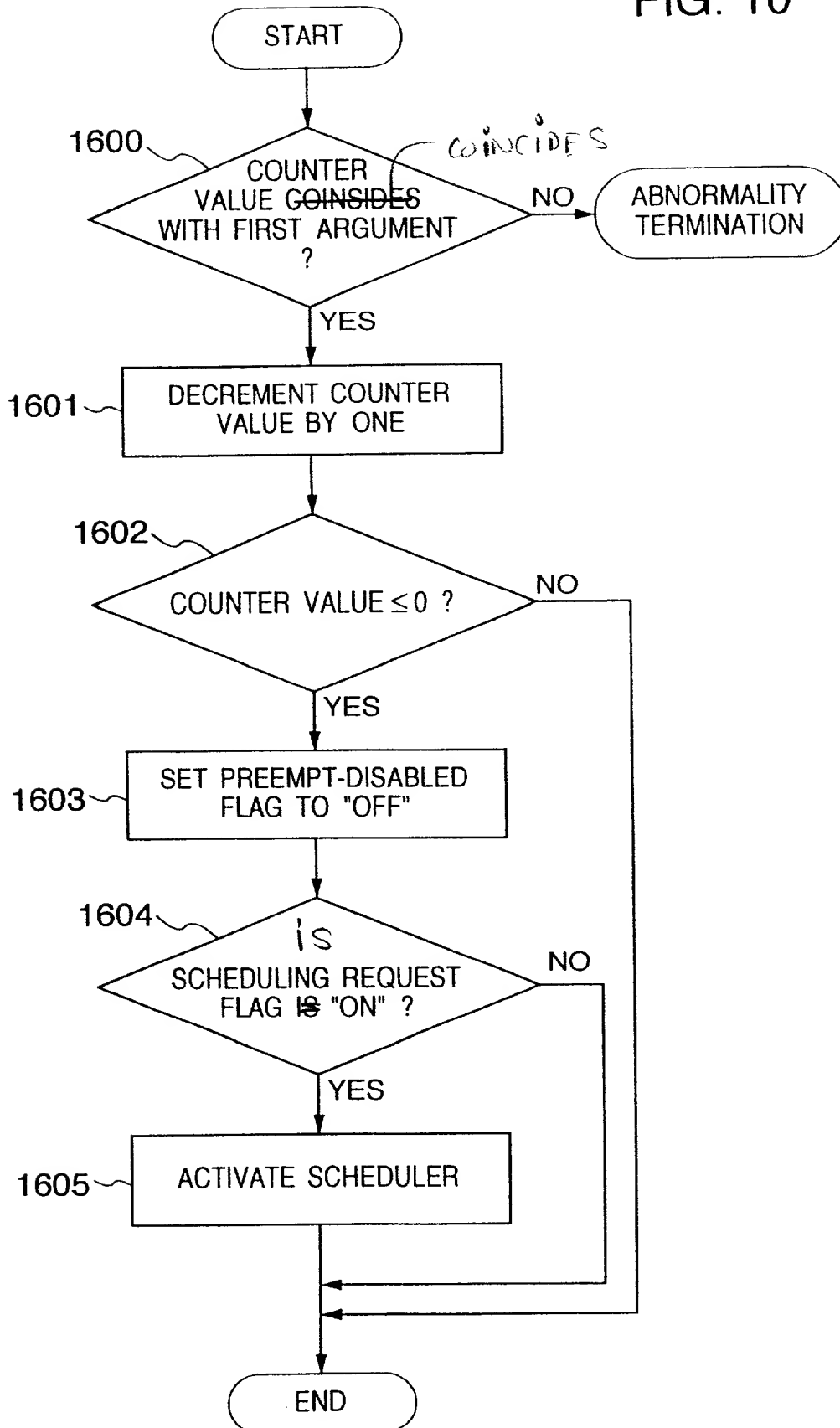
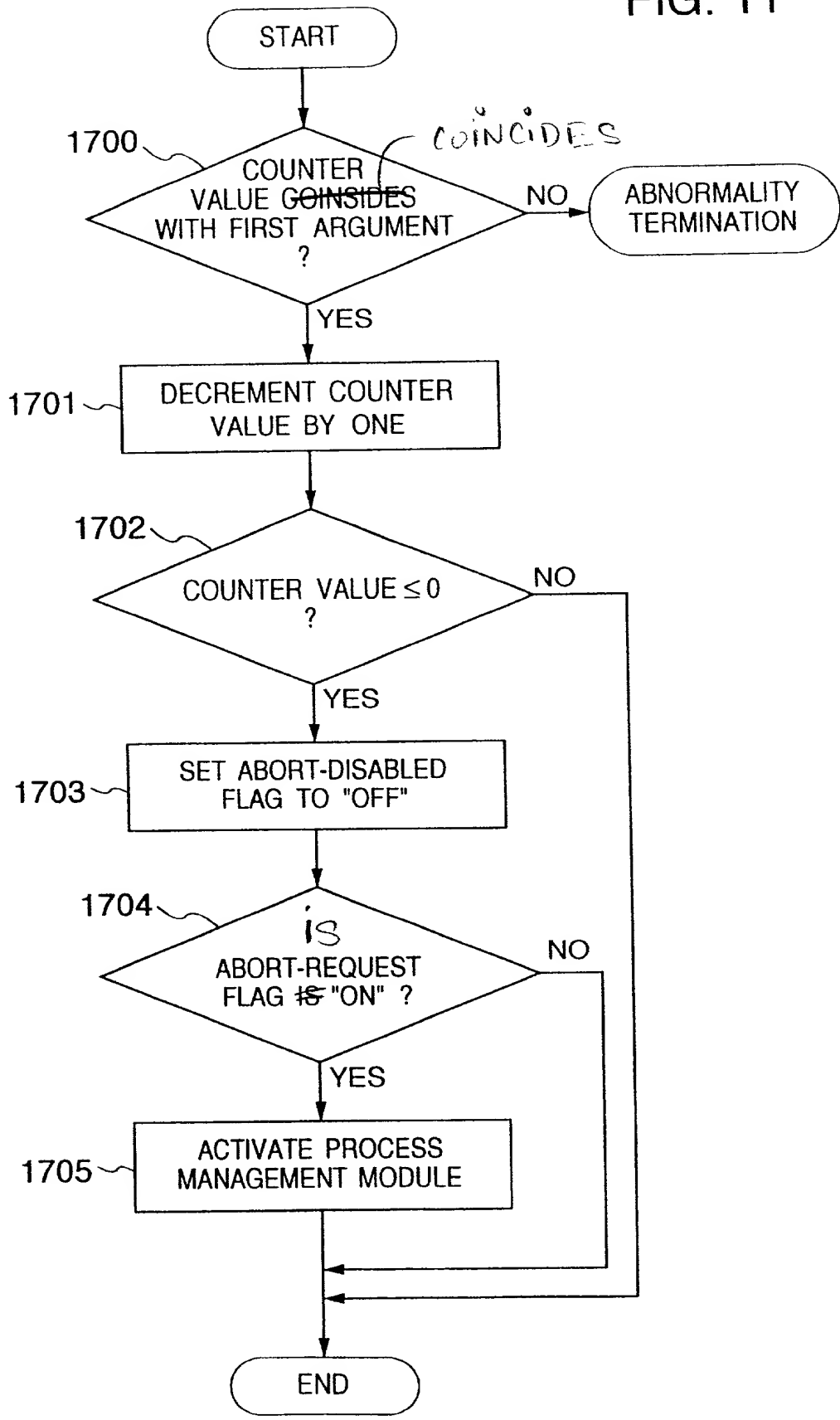


FIG. 11



09337-04200  
T00240-093360



FIG. 13

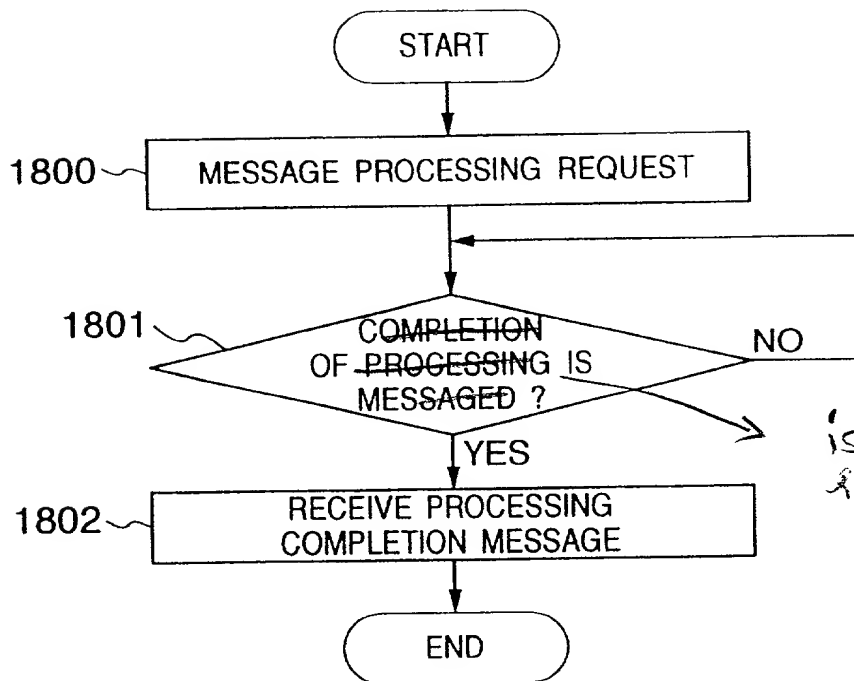


FIG. 14

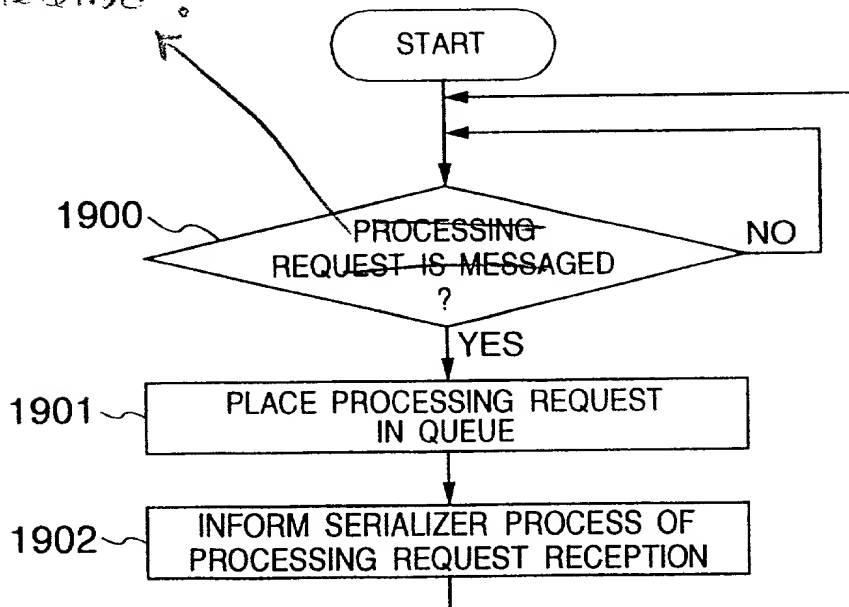


FIG. 15

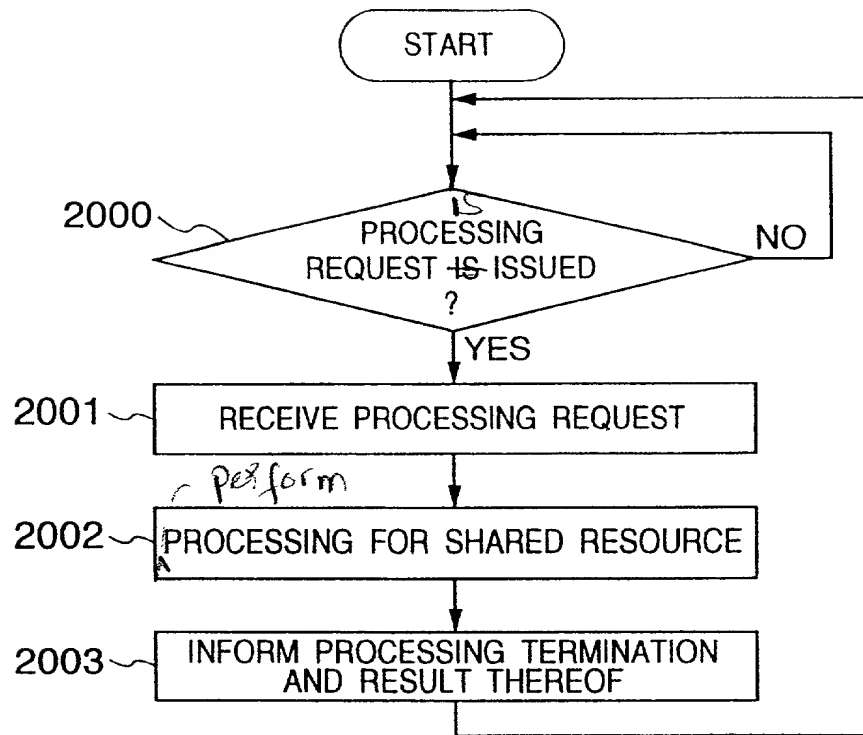


FIG. 16

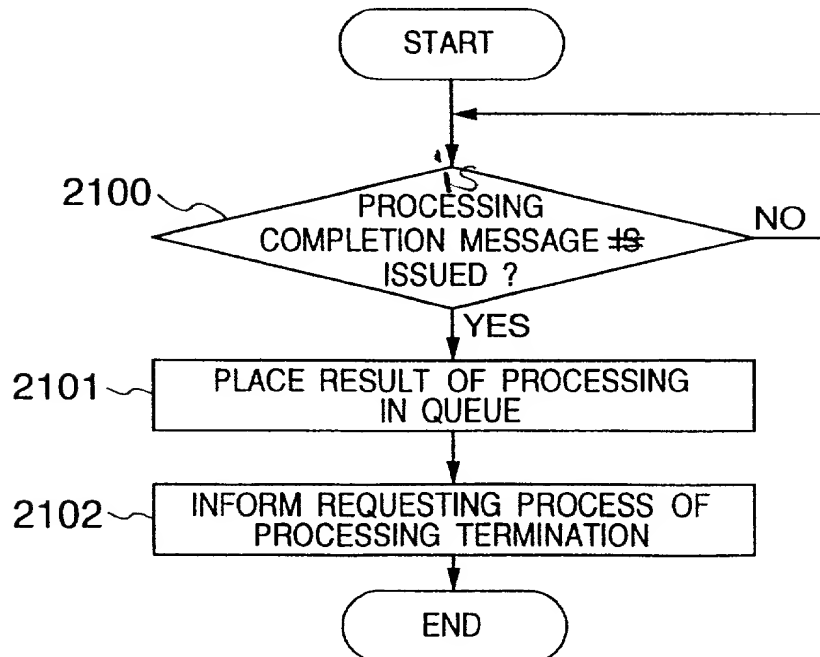


FIG. 18

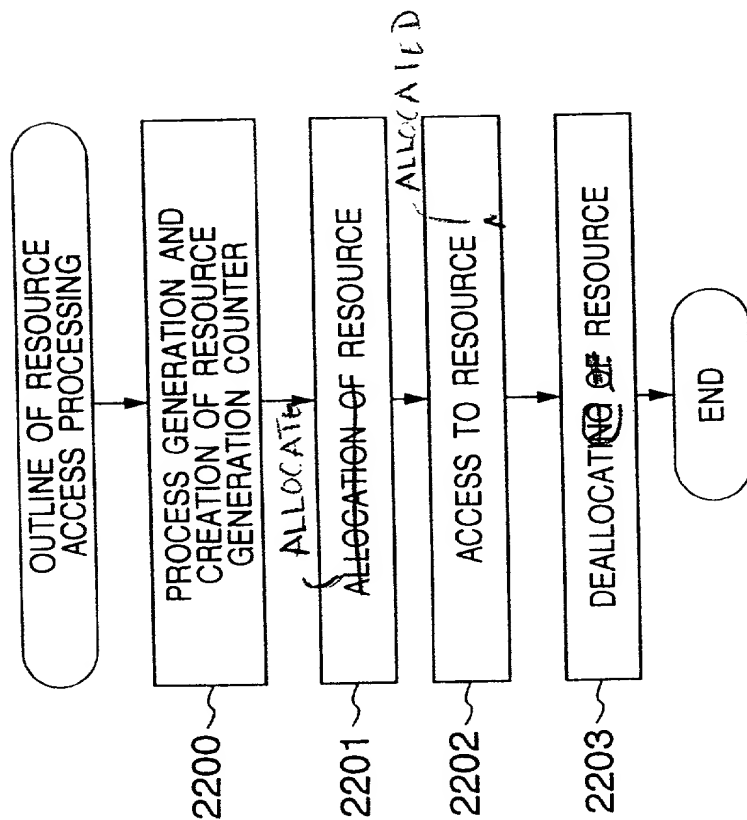


FIG. 19

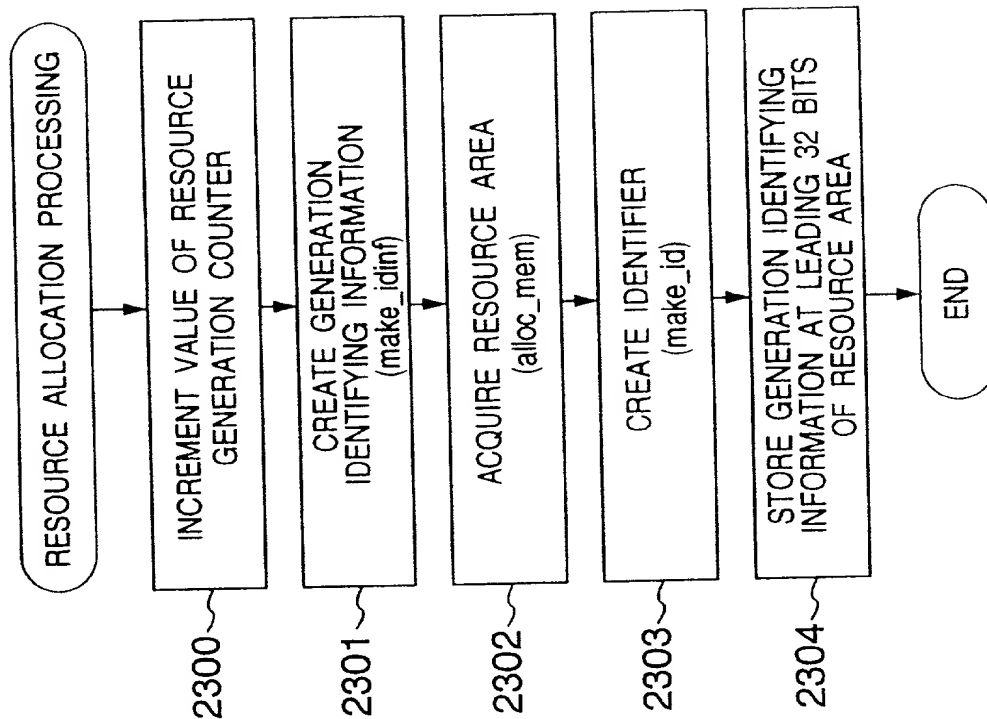


FIG.24

